

# CONNECTED

From the Commodore's Blue Screen  
to Artificial Intelligence



**LUCIO DAMIAN MOSCHELLA**

# **CONNECTED**

From the Blue Screen of the Commodore to Artificial Intelligence

Lucio Damián Moschella

2026

To my father. Who never taught me the future. He simply put it in front of me.

© Lucio Damián Moschella, 2026

First digital edition

Argentina / Mexico

All rights reserved.

This work is shared free of charge exclusively for personal reading and personal use. Its reproduction, distribution, translation, adaptation, full or partial publication, or any other unauthorized use, by any means, is prohibited without the author's prior written consent.

All trademarks and registered trademarks mentioned in this book are the property of their respective owners.

For contact and permissions:

LinkedIn: <https://www.linkedin.com/in/lucio-damian-moschella-23b1b9a/>

# **Table of Contents**

Prologue — Before We Begin

## **ACT I — The Box That Hummed**

Chapter 1: The Inheritance

Chapter 2: The Language of Machines

Chapter 3: The Tribe Without a Name

## **ACT II — The Map No One Finished**

Chapter 4: The Page That Changed Everything

Chapter 5: The Technician in the Storm

Chapter 6: When Chaos Found Its Order

## **ACT III — The Frontier That Breathes**

Chapter 7: The Second Threshold

Chapter 8: Where Control Ends

Chapter 9: The Cursor Still Blinks

About the Author

## Prologue — Before We Begin

I'm not the smartest person in the room. I never felt that I was. But a long time ago I learned that this is not the only way to stand close to something important.

What I am is someone who was there.

I was there when a computer was still a rarity in a home. When the internet was not yet part of the landscape, but more like a blurry promise. And I am here now, in another moment that feels strangely similar, watching a new technology emerge that brings back that same mix of wonder, confusion, and vertigo I had already felt before.

I'm not writing this book from an academic perspective. Nor from the position of someone who believes he has definitive answers. I'm writing it from experience. From having seen, twice, how a technology begins as something nearly incomprehensible to most people and, little by little, ends up changing everyone's lives.

That does not make me a guru or a prophet. It makes me a witness.

And with time, you learn that being a witness also means being compelled to get involved. Not because you have to preach or teach lessons, but because after seeing certain things up close, remaining completely silent would also be a way of wasting that experience.

This book comes out of that.



It also feels important to say early on what this book is and what it is not.

It is not a technical manual. You will not find tutorials, recipes, or instructions here for using artificial intelligence tools. Nor will you find a step-by-step guide for implementing anything. There are people far better prepared than I am to write that kind of book, and they do it better too.

What I wanted to do here was something else. Something less exact, perhaps, but more human.

I wanted to place this moment in perspective. To try to look at it not just as a novelty, but as part of a longer story. A story made of machines, yes, but also of people. Of curiosity. Of mistakes. Of intuition. Of excessive enthusiasm. Of fear. Of people learning while things were already happening.

Because that is how these revolutions happen. They never wait for us to fully understand them.



There is something about the current conversation around artificial intelligence that makes me uneasy.

On one side, there is euphoria: the sense that this is going to fix everything, simplify everything, inaugurate an almost perfect era. On the other side, there is fear: that it will destroy jobs, concentrate power, make many people irrelevant, and change the rules too fast and too badly.

And the truth is, both reactions make sense to me.

What strikes me is something else: the lack of memory.

Not because the past gives us ready-made answers, but because it helps keep us from talking as if all this had fallen from the sky. We have lived through something similar before. Not the same, of course. But similar in atmosphere, in speed, in disorientation, in that feeling that something enormous was beginning and no one could yet fully grasp its consequences.

That happened with the internet.

There were exaggerated promises then too. There was genuine fear then too. There were people who saw before others what might be coming, and there were also many who thought it was just another fad, something peripheral, something they could watch from a distance until it became clearer.

By then, it was already too late to watch from a distance.



I don't know exactly where artificial intelligence is headed. And honestly, I don't think anyone truly knows.

We can sense it, project it, speculate about it. But know it, really know it? No. I deeply distrust anyone who speaks about this with total certainty, as if describing something already settled.

What I do know is something else: staying out of it while waiting for total clarity almost never ends well.

The technologies that truly change the world tend to arrive this way. First they unsettle. First they confuse. First they generate noise. Later, with time, come the explanations, the rules, the categories, the manuals. But by then the change has already begun.

I have seen that happen too.

And I think that is why I felt the need to write this book: not to say "I know," but to say, "This has felt this way before, and it is worth paying attention."



My father was an electronics engineer.

He was not a man of grand speeches. He never sat me down to explain a theory of the future, nor did he talk to me about technological revolutions or anything resembling that. He did something much simpler.

One day he brought a computer home, put it in front of me, and let whatever was going to happen happen.

I was nine years old.

I remember the blue screen. I remember the blinking cursor. I remember the silence of the dining room at night, that strange light from a television connected to a machine he had repaired with his own hands. I didn't really understand what I was seeing. Not even close. But I did feel there was something there. Something that called to me in a different way.

Over the years, I understood that my father had not given me only a computer. He had placed in front of me a new way of looking at the world.

And he trusted that curiosity would do the rest.

He was right.



This book also begins there.

With that rather simple idea that sometimes you do not need to explain everything to someone. Sometimes it is enough to show them something true and let them move toward it. Look at it. See what it does to them.

That is what I wanted to do here.

Not to explain the future as if I could. Just to place it in front of you. To try to trace a line between that blue Commodore 64 screen and this present in which we are speaking to machines again, only now they no longer limit themselves to following instructions: they respond, interpret, propose, and sometimes even seem to anticipate.

That still amazes me. And to be honest, I still struggle with it a little. Not technically, but humanly. In trying to understand what it really means to be living through this now.

Maybe that is why I wrote.

To make some sense of what I saw. To leave a trace of that experience. And also to speak to anyone who may be feeling something like what I felt back then in front of that first screen: the intuition that something important is happening, even if you still cannot quite put it into words.



It is still early.

There are still more questions than answers. There is still a great deal of noise. A great deal of exaggeration, in both directions. But precisely because of that, it seems to me that this is the moment to look closely.

Not when everything has already been settled. Not when the map has already been drawn, bound, and explained by others. Now. In this uncomfortable stage. In this part where you still have to pay real attention.

Because later, things normalize. They become part of the landscape. And you forget that once they were extraordinary.

I have not forgotten that first time.

I still see that blue screen when I close my eyes. I still hear the sound of old machines. And somehow, even though decades have passed and everything seems so different, I still feel the same thing whenever something genuinely new appears: that mix of curiosity and respect. Of enthusiasm and caution. Of not fully understanding, and still wanting to move closer.

The cursor blinks.

And deep down, I think it is still waiting for the same thing from us: that we dare to begin.

## **ACT I — THE BOX THAT HUMMED**

## Chapter 1 — The Inheritance

My father had the hands of an engineer. Hands that knew what they were touching.

When he brought home the computer, it did not arrive in a brand-new box. It came the way truly important things come: without packaging, a little worn, with a history of its own. A friend of his, who had a computer company, had gotten hold of a Commodore 64 that nobody had been able to make work. Something with the keyboard. A problem that, for most people, meant junk, but for an electronics engineer meant something else: an opportunity.

My father bought it like that. Broken.

I don't know how many days it sat on his workbench. I watched him leaning over it with that quiet concentration he had whenever something really mattered to him, with tools I didn't know how to name, with the patience of someone who understands that machines do not break out of malice: they break because something stopped connecting to something.

He repaired it.

And when he set it up in our dining room, connected to the television with a cable that seemed far too simple for everything it promised, there were already five diskettes beside it. Five diskettes filled with programs and games someone had gotten hold of before the machine even worked. We still didn't have the disk drive to read them, but they were there anyway, like a promise. As if the future had arrived a little before the way to open it.

The screen turned blue. The cursor blinked.

I was nine years old and I didn't yet understand that I was looking at the beginning of something. I only knew that light called to me in a way television never had. Television gave you things. This asked something of you. It asked you to speak.

My father typed something, and the screen responded.

Some time later he bought the 1541 disk drive, that beige block almost the same size as the computer itself, with its mechanical read head searching for data as if it were thinking out loud. Only then did those five diskettes stop being a promise and begin to open for real. We spent hours loading programs, waiting for that combination of

squeaks and clicks that meant something was about to appear on the screen. I learned, before I understood it, that waiting was also part of the process. That patience was a technical skill.

What my father gave me that afternoon was not a computer.

He gave me a language.

A language that machines spoke and that very few people in the world knew yet. A language he had already decided I was going to learn, not because he explained it to me, but because he placed the tool in front of me and trusted that curiosity would do the rest.

He was right.

Thirty-five years later, in front of a very different screen, I wrote something to an artificial intelligence in natural language — not in code, not in commands, in words — and it answered with something I did not expect. Something that went beyond what I had asked. As if it understood not only the instruction, but the intention behind it.

I thought of my father. Of the broken machine nobody wanted. Of his hands repairing it in silence.

That some people do not teach you the future. They place it in front of you and trust that you will know what to do.



*In 1983, while that cursor blinked in the dining room of some ordinary home, in garages in California, in university basements in Massachusetts, in apartments in Tokyo and London, other cursors were doing the same thing. Nobody coordinated it. Nobody planned it. It was simply the moment: that specific instant in history when a technology stops being an experiment and begins to look like a promise. A promise that still had no name, but was already alive in five diskettes stacked beside a lit television.*

*That happens twice in a life, if you are lucky. I was lucky twice.*

## Chapter 2 — The Language of Machines

The manual smelled like new paper and had a cardboard cover. It was thicker than something that mysterious should have been.

I would open it to any page and find things I didn't understand. Variables. Loops. Commands in English that nobody had taught me because nobody around me knew them either. The technical English of the 1980s was not the English of school. It was another language inside the language — a tongue that lived exclusively between the covers of manuals that came in the boxes of machines very few people owned.

So I learned the way you learn when you have no other choice: by copying.

Magazines came with programs printed in them. Entire pages of BASIC code in typewriter font, with line numbers that started at 10 and advanced by tens — 10, 20, 30, 40 — as if the program needed a little room to breathe between instructions. You would place the magazine next to the keyboard and start transcribing.

```
10 PRINT "ENTER YOUR NAME: "  
20 INPUT N$  
30 PRINT "HELLO, ";N$  
40 GOTO 10
```

Line by line. Character by character. Without yet understanding why it worked, only trusting that if you copied it correctly, something would happen on the screen.

And when it worked — when the screen responded exactly the way the magazine promised — you felt something that had no exact name. It was not only satisfaction. It was the first time you understood that you had done that. That between your hand, the keyboard, and the machine, something had happened that did not exist before.

When it did not work, the most important part began: finding the error.

One wrong character. One missing quotation mark. One miscopied line number. The C64 gave you back a message without mercy and with very little explanation. SYNTAX ERROR IN LINE 30 was all you got. The rest was on you. You had to go back to the paper, compare each symbol, train your eye to see the difference between an O and a zero, between a 1 and a lowercase l.

Nobody taught you how to debug. You learned because there was no other option.

That too, though I didn't know it then, was programming. Not the glamorous part you see in movies — not screens full of green code cascading downward — but the real part: patience, attention to detail, the ability to hold an entire system in your head while you looked for the exact point where something stopped connecting to something.

Information was scarce and physical. If you got a new magazine, it was an event. If someone handed you a photocopied manual, you treated it like something valuable. There were no forums. There was nobody on the other side of a screen ready to answer you in seconds.

There was paper. There was time. There was a blue screen waiting.

And yet — or perhaps precisely because of that — what we learned, we learned for real. Because every line of code you understood, you had earned. Nobody had explained it to you in an eight-minute video. You had deciphered it yourself, with the manual, with the error, with the patient comparison between what the paper said and what the machine said.

Forty years later, a teenager opens a chat window and asks an artificial intelligence how to make a loop in Python. In seconds, they receive not only the code, but the explanation, the use cases, the common mistakes, and three alternative versions. Access to knowledge multiplied in a way that, in 1983, was literally unimaginable.

But something still nags at me when I think about that.

I don't know whether that teenager will remember the loop tomorrow.

I remember GOTO 10 forty years later. I remember it because it cost me something. Because it was written on paper before it was on the screen. Because when it worked, I was the one who made it work.

Knowledge gained through friction settles differently from knowledge that arrives without resistance. I'm not saying friction is better — I'm saying it produces something different. And that in 2026, as we rightly celebrate the democratization of knowledge, it is worth asking what we lose when learning stops costing us something.



*Computer magazines of the 1980s were the only network that existed.  
Compute!, RUN, Commodore User, Ahoy! — publications that circulated*

*among the few who spoke that strange language of machines. In their pages lived programs, yes, but also letters from readers in which someone in Nebraska would ask a question and someone in London would answer two months later in the next issue. It was slow, incomplete, and gloriously human. The first internet had no cables. It had paper and postal time.*

## Chapter 3 — The Tribe Without a Name

Before the internet, there was a sound.

It was not music. It was not exactly noise. It was something in between — a sequence of squeals, tones, and static that lasted twenty to thirty seconds and that, if you recognized it, meant you were one of us.

It was the modem connecting.

The process was a ritual you learned by heart. First you dialed a phone number — a real, local number someone had passed along on paper or one you had found in a photocopied list circulating among the few who knew such lists existed. There was no public directory. There was no place to search. There was an invisible network of people passing information in person, hand to hand, as if it were a secret that wasn't really a secret — only something invisible to anyone not looking for it.

Then you waited. The line rang. And then the sound began: the greeting of two modems recognizing each other, negotiating their speed, agreeing in fractions of a second on the language they were going to speak. Three hundred baud. Six hundred. Twelve hundred if you were lucky and the machine on the other end could handle it.

Speeds that today, as numbers, barely mean anything. But back then they were the difference between connected and disconnected. Between inside and outside.

When the screen finally showed text — the greeting of a BBS, the menu of options, the name of the system receiving you — you felt something that still had no precise name. Years later I would learn to call it presence. The certainty that there was someone on the other end. That the machine was not an island.

BBSs were electronic bulletin boards. Bulletin Board Systems. Someone — usually an enthusiast with a computer, a modem, and a dedicated phone line — would set up a system that others could call into, read messages on, leave their own messages on, and download files from. It was almost always asynchronous: you left your message and came back the next day to see whether someone had replied. The conversation did not happen in real time. It moved more slowly. More deliberately.

But sometimes — in direct connections between two machines — you could chat in real time.

I did it with a friend who had the same setup. Commodore 64, modem, and the desire to see whether all of it really worked. One person would call the other. The modems would go through their recognition ritual. And then, on separate screens in separate houses, we would begin writing to each other.

One letter at a time. Because that was how it worked — every character you typed appeared instantly on the other person's screen. Without waiting to finish the sentence. Without deleting. Everything visible, everything live, everything imperfect.

It was the slowest and most fascinating conversation I had ever had in my life.

There were no emojis. No way to know whether the other person was thinking or had gone to the kitchen to get something. Just the blinking cursor, waiting for the next letter. You typed more carefully because there was no way to edit what had already appeared on the other person's screen. Every word cost something. Every message was a decision.

We were a tribe without a name.

We did not necessarily know each other in person. What united us was something more specific and stranger: access to a frequency the rest of the world still was not tuned to. We knew things our neighbors did not know. We knew computers could talk to each other. That distance could collapse into a phone line and a pair of modems negotiating their language through thirty seconds of noise.

The world did not know we existed. And that, strangely enough, made us feel more connected. We shared something that could not be easily explained — because to explain it, you needed to have been there, to have heard the sound, to have watched text appear on a screen as if it were coming from another planet when in fact it was coming from the other side of town.

What we did not know was that we were rehearsing something.

That those BBSs with their text menus, those photocopied lists of phone numbers, those letter-by-letter chats between two Commodores — all of that was the draft of something that still did not have a final shape. A network that still did not call itself the internet. A community that still did not call itself online. A revolution that still did not know it was a revolution.

We were the generation that arrived before the map.

And in 2026, when I watch someone interact for the first time with an AI agent — that particular look of someone who has just discovered there is something on the other end that answers, that understands, that continues the conversation as if it had been waiting for exactly that question — I recognize something I thought I had left behind thirty years ago.

It is the same look.

The same exact moment when the machine stops being an object and becomes a presence. The same mixture of fascination and vertigo. The same question, still unanswered: How far does this go?

In 1987, we had no answer. In 2026, we do not have one either. And that absence of an answer — that territory without a map — is not a flaw in the system. It is, rather, the sign that something genuinely new is happening.

The tribe does not have the same name now. In 2026 that group no longer fits inside the word tribe — it is a tide. But those who have already crossed that threshold, those who truly recognize what is happening, still recognize each other with that same look from back then. The same moment before the map, before the answers, before someone writes the manual. Only this time — unlike then — we know that manual is never going to arrive in time.



*BBSs disappeared when the internet made them unnecessary. But for nearly a decade they were the only network there was — invisible to most, vital to the few who knew which number to dial. At their peak, in the early 1990s, there were an estimated sixty thousand active BBSs in the United States alone. Sixty thousand systems running on personal computers, in basements and teenagers' bedrooms, silently connecting a tribe the world did not yet know existed.*

## **ACT II — THE MAP NO ONE FINISHED**

## Chapter 4 — The Page That Changed Everything

My father arrived one day with the same calm he always had.

He did not announce anything. He did not prepare the moment. He simply said there was something he wanted to show me and that to see it, we had to go out. It was not at home. It was not in any manual. It was in a little shop a few blocks away — a place someone had set up with a few computers connected to something that very few people yet knew how to name precisely. The owner was an engineer. Like my father. He had managed to get a connection and had decided other people needed to see it.

I sat down in front of the screen.

I don't remember if it was Mosaic or Netscape. Mosaic had been the first to display images inside a web page, the first to give the internet something resembling a face. Netscape took all of that and made it faster, more stable, more like something an ordinary person could use without a manual.

The engineer typed an address into the bar — a sequence of letters, dots, and slashes that looked like nothing I had seen before — and pressed Enter.

The screen took its time.

Not seconds. More than that. The modem was negotiating with something somewhere else in the world, asking for information, waiting for it to arrive in fragments through telephone lines that had not been designed for this. Every second of waiting was technology working at the limit of what it could do.

And then it appeared.

A page. Text. An image loading from top to bottom, line by line, as if it were being drawn in real time. Nothing spectacular if you describe it in words. Completely impossible to describe in words if you want to convey what it felt like to see it.

Because what was on that screen was not just a page. It was an address in a space with no physical geography, not printed on any map, existing on some server somewhere in the world I could not point to with my finger. Someone had built it. Someone had decided to place something there, in a space that did not exist until someone filled it, and now I was seeing it from a shop in my city.

Distance had disappeared.

Not all at once. Not dramatically. Silently, the way things disappear when they are going to change everything.



What I did not know then was that there were books.

Physical books, printed on paper, trying to explain and organize that new world. The most famous was *The Whole Internet User's Guide* by Ed Krol, published in 1992. It was not just a list of URLs — it was the instruction manual for an entire world. It explained what email was and how to use it, how to connect to remote computers through telnet, how to transfer files through FTP, how to search for information with tools like Archie or WAIS, which almost nobody remembers now but which back then were the closest thing that existed to a search engine. It was a map of a territory that was changing even as they drew it.

They were the Yellow Pages of the internet.

Think about that for a moment. Someone sat down, visited websites one by one, classified them, wrote descriptions of each one, sent the manuscript to a publisher, waited through the publishing process, and by the time the book reached the shelves, several of those sites already no longer existed and hundreds of new ones had appeared that the book could never include.

It was absurd. It was inevitable. It was exactly what people do when something new appears and the infrastructure to navigate it does not yet exist: you translate it into the format you already know. You put it on paper. You give a familiar shape to something that does not have one yet.

Yahoo! appeared in 1995 with the same logic, but digitally — it was not an automatic search engine but a human directory. Jerry Yang and David Filo, two Stanford students, categorized websites by hand and organized them into sections. Real people visiting each site and deciding where it belonged. The algorithm came later. First came human beings trying to impose order on something growing faster than any effort to organize it.

And GeoCities — launched in 1994 — was the first time the internet gave ordinary people a voice. Anyone could create their own web page for free. No programming

required. No server required. The pages were visually awful — black backgrounds, blinking yellow text, visitor counters, animated GIFs in every corner. They were also completely revolutionary. For the first time in history, any person with access to a connected computer could publish something the entire world could read.

The threshold had dropped. What once required a university server and technical knowledge could now be done by almost anyone with enough desire.



In 2026 there are directories of artificial intelligence agents. Lists of tools organized by category, by function, by price, by degree of complexity. Sites trying to capture, in one frozen moment, something that changes every week. New models appearing. Tools disappearing. Entire categories that did not exist three months earlier.

They are the Yellow Pages of AI.

But there is something else happening in 2026 that has no exact equivalent in the 1990s — something that changes the rules in a way we are still trying to process. In the internet era, tools were passive. A web page waited for you to go looking for it. An email arrived and waited for you to open it. The web was a vast and extraordinary repository, but inert. You navigated. You searched. You decided every step.

AI agents do not wait.

An agent is not a tool you answer back to — it is something that acts. You describe a goal in natural language, and it decides how to move toward it: it searches for information, drafts documents, sends messages, executes code, coordinates with other agents, makes intermediate decisions without asking your permission at every step. Agentification — that new word beginning to appear in nearly every serious conversation about AI in 2026 — is the moment when technology stops being an instrument you touch and becomes something that works while you sleep.

That has no precedent in the history of the internet.

And yet democratization bears the same familiar signature. GeoCities gave anyone the ability to exist online without knowing how to code. Today, anyone can create an AI agent with a natural-language description without fully understanding models or code.

Access opens up. The threshold drops. And the world that seemed reserved for technical specialists suddenly becomes available to everyone.

The maps were never really about the territory. They were about us — about needing something to hold onto while the ground shifted.



That afternoon, my father's engineer friend showed us a few more pages. Each one was a small miracle of the same thing — text and images arriving from somewhere far away, slow, fragmented, but real. Present.

Before we left, my father asked him something. I do not remember the exact words, but I do remember the tone — the tone of someone who already understands what he is seeing and wants to confirm that the other person understands it too.

The engineer nodded.

Two men who had spent their careers learning the language of machines, standing in an almost empty shop in some ordinary city, looking at the beginning of something that was going to change everything. Unable to say exactly how. Unable to say exactly when. Holding only that calm certainty — the certainty of engineers — that what they had just seen had no turning back.

They were right.

Those who arrive before the noise are always right.



*In 1993 there were just over a hundred websites in the entire world. In 1994 there were ten thousand. In 1995 there were a hundred thousand. By 2000 there were hundreds of millions. The growth was not linear: it was an explosion no model predicted correctly, no directory could capture, no analyst could fully measure before it happened. The internet did not simply grow. It detonated. In 2026, AI models are multiplying at a similar pace. Those of us who lived through both explosions know how to recognize the*

*pattern. And we know, too, that when something truly detonates, the maps always arrive late.*

## Chapter 5 — The Technician in the Storm

There is a difference between seeing a revolution and living it from the inside.

The person who sees it from the outside observes the phenomenon, reads the headlines, maybe buys stock at the wrong moment. The person who lives it from the inside gets to the office early because the server crashed at three in the morning, spends hours on the floor of a company that smells of new carpet and cigarettes, and explains to one manager who does not understand why they need an IP address the same thing he explained to the previous manager the week before and to the next one the week after.

I was inside.

In the mid-1990s, while I was studying systems and working in technology, the corporate world was discovering the internet with the same mix of excitement and confusion with which a child discovers he can start a fire. They knew it was powerful. They did not know exactly what to do with it. And they needed someone who knew — or at least knew more than they did — to tell them what to buy, what to connect, and how to make it work.

Networks were the first problem.

Before any company could think about the internet, it had to solve something more basic: connecting its own computers to each other. Novell NetWare was the dominant network operating system of that era — software that made it possible to share files and printers between computers, with its own logic, its own protocol, its own way of thinking about permissions and users. Learning Novell meant learning an entire language that had nothing to do with DOS or Windows. It was technical, dry, and exactly what companies needed without yet knowing they needed it.

Then Linux arrived. Free, powerful, built by a global community of programmers who shared code instead of selling it — an idea that, in the 1990s, sounded almost subversive from an economic point of view. An operating system that belonged to no one and could be modified by anyone. One you could run on old hardware that otherwise would have been junk. One that gave you total control over every aspect of the server, if you knew where to look.

And then came Windows Server — which simplified all of that at the cost of taking away some control. Easier to sell. Easier to explain. Easier for a manager to approve in a budget without understanding too many technical details.

That was the real work of that period. Not glamorous. Nothing like what the movies showed. Cables, configurations, passwords nobody remembered, updates that broke things that were already working, and the constant feeling of building the airplane while flying it.



But underneath all of that — beneath the Novell servers, the network cables, and the printers that never worked on the first try — something was growing at a speed no technical manual could anticipate.

The internet arrived in companies as a question before it became an answer.

Do we need a website? That was the question in 1996. It was asked by managers who did not know what a server was, directors who had read something in a business magazine, business owners who had seen the Netscape logo on their son's computer and vaguely connected it to some opportunity they could not quite define.

The honest answer was: I don't know whether you need one, but if you don't have one in two years, you're going to look invisible. Nobody gave that answer. The answer people actually gave was: yes, you need one, and I can build it for you.

And that is how a generation of technicians became, almost without meaning to, consultants, salespeople, and translators between two worlds that did not speak the same language. The world of people who understood technology and the world of people who had to use it without understanding it. We were the bridge. Underpaid, overloaded, and absolutely indispensable.

The ironic thing is that we did not fully know where any of this was headed either.

We knew how to configure an Apache server on Linux. We knew how to build a network with Novell. We knew how to install Netscape on twenty computers in a single morning. But if someone had asked us in 1997 whether people would be buying everything online in ten years, whether something called social media would exist where millions would post pictures of their lunch, whether phones would become

pocket computers with permanent connections — most of us would have answered with a mixture of skepticism and hope that, in retrospect, seems completely reasonable.

Nobody knew. The ones who claimed they did were lying or fooling themselves.



The dot-com boom arrived with money that made no sense.

Companies that had sold nothing, that had no clear business model, that existed only as a URL and a vague promise of a digital future, were receiving millions in investment. The market was betting not on what those companies were, but on what the world might become if the internet grew in the right direction. It was a bet on the territory, not on the map.

Some of those bets were right. Amazon had existed since 1995 and lost money for years before becoming what it is today. Google appeared in 1998 and took time to find its business model. The seeds of what is now the digital world were planted in that period of irrational euphoria — amid companies that collapsed and left behind nothing but a remembered logo and an expired domain.



In 2026, the pattern repeats itself with an unsettling precision.

On November 30, 2022, OpenAI released ChatGPT. In five days it had one million users. In two months, one hundred million. No technological product in history had ever grown that fast — not even the internet in its best boom years. It was not only the product itself. It was the public demonstration, available to anyone with a computer and an internet connection, that something had changed irreversibly. That machines could converse. That they could reason. That they could produce in seconds what had once taken hours.

The world was not prepared. And, strangely enough, that was exactly what should have been expected.

Anthropic — founded with the explicit goal of developing reliable and safe AI — built Claude, a model that not only converses but can use external tools, execute code, and operate computers autonomously. Google responded with Gemini, integrated into its product ecosystem and connected to real-time world data. Meta released its LLaMA models — not exactly open source in the strictest sense, but openly available enough for companies and researchers to download, modify, and deploy on their own servers. And Microsoft, after bringing OpenAI into the center of its strategy, built Copilot — AI inside Windows, inside Office, inside Teams — and layered an enterprise stack around it to connect models with real data through Azure, Copilot Studio, and the kinds of connectors that do what Novell did in the 1990s: create invisible order beneath something users rely on without ever thinking about the infrastructure.

Five companies. Five different bets on the same territory. The same chaos as 1996, only with more zeroes in the valuations and gigabytes instead of kilobytes.

And beneath the noise — just as before — are the technicians. The ones learning LangChain and n8n to orchestrate agents. The ones configuring AI pipelines inside companies that still do not fully know why they want them. The ones arriving early because the inference model failed at three in the morning. The ones explaining to a manager what an AI agent is with the same patience they used in 1997 to explain what an IP address was.

The bridge between two worlds. Again.

The question being asked in boardrooms in 2026 is, at heart, the same one that was being asked in 1996 — only one word has changed: Do we need AI? And the honest answer almost nobody says out loud is this: I do not yet know exactly how you need it, but the train has already left the station. The question is no longer whether to get on. It is whether you still have time to catch it.

The companies that, in 2026, are still waiting for someone to explain the complete manual before moving are going to discover too late that no such manual exists, that the market did not wait, and that while they were analyzing, others were already operating differently. That is not a threat. It is the same pattern as always. In 1996, the companies that ignored the internet did not all disappear at once — they simply began becoming invisible, slowly, until one day they looked around and the world had moved on without them.

I know that feeling from both sides of the desk.

I lived it while configuring Novell servers in companies that did not really know why they wanted them. I am living it again while implementing AI in organizations that still do not quite know what they are going to do with it.

The technician in the storm. Again.



*The dot-com crash of 2001 wiped thousands of companies off the map — companies that had promised to change the world. But it did not erase the internet. What survived was not the loudest part, but the most solid part: the infrastructure, the protocols, the habits people had already absorbed and were never going to give up. Crises do not destroy revolutions. They clean them out. They separate the noise from what truly changed forever. In 2026, as investment in AI is moving faster than dot-com investment did at the height of that euphoria, it is worth asking not what will explode, but what will remain once the noise settles. The answer, as always, will be in the infrastructure. In the tools nobody names in the headlines but everyone will use without even knowing it.*

## Chapter 6 — When Chaos Found Its Order

There was an exact moment when the internet stopped being a wild territory.

It was not an announcement. It was not a law. It was an algorithm.

In 1998, two Stanford PhD students named Larry Page and Sergey Brin launched a search engine with a name that sounded like a typo — Google — and forever changed the relationship between human beings and information. The idea was simple in its statement and revolutionary in its execution: instead of classifying web pages by their content, they ranked them according to how many other pages cited them. Popularity as a signal of relevance. The network voting for itself.

Before Google, searching for something on the internet was an adventure with no guarantees.

AltaVista could give you useful results or completely irrelevant ones. Yahoo was a human directory that depended on whether someone had manually classified what you were looking for. Searching for information on the internet in 1997 felt a lot like looking for a book in a library where nobody had finished organizing the shelves — you knew it was there somewhere, but finding it depended on your patience and a little luck.

Google solved that. All at once, without warning, with a white search box on an almost empty page that contrasted with the visual saturation of nearly everything else online at the time.

You typed something. You hit Enter. And there it was.

What nobody measured at the time — because it was impossible to measure while it was happening — was the side effect of that revolution. Google did not only organize the internet. It normalized it. It made it ordinary. It turned it into invisible infrastructure — like water flowing from a faucet without your thinking about the pipes, like electricity reaching a socket without your remembering the power plant.

At some point between 1998 and 2005, the internet stopped being a novelty and became the air.

And when something becomes the air, you stop marveling at it.



I noticed it in conversations.

In 1996, explaining the internet to someone was an event. There were questions, there was amazement, there was that specific look of someone processing something that was changing their model of the world. By 2003, the same explanation produced a polite nod and a change of subject. Not because the technology had stopped being extraordinary, but because it had stopped being new. And what is new is what activates wonder.

The generation born after Google never knew the internet as a wild territory. For them, there had always been a search engine that worked, always ordered results, always an answer one click away. They never lived the thrill of finding something valuable in chaos because, from their perspective, there had never been chaos. There was only information — available, indexed, accessible.

What they gained in access, they lost in wonder.

That is not a criticism. It is an observation about how technological revolutions work. There is always a generation that lives the before and after — that remembers the sound of a modem and also knows broadband, that used AltaVista and then used Google, that built networks with Novell and later deployed services in the cloud. That generation has something later generations cannot have: the perspective of contrast.

It knows what things used to cost. And because of that, it understands what they are worth now.



In 2026, we are exactly in the moment before the Google of AI.

The territory is still wild. Models multiply. Interfaces change every few months. A shared language does not yet exist. On one side are the tools for building agents — LangChain, AutoGen, CrewAI, n8n — powerful, flexible, but still rough and technical, designed mostly for people who have already learned how to think in prompts, flows, tools, and memory. Frameworks that give you the power to build almost anything, but require you to know exactly what you want to build and how.

On the other side, agents that come ready to work are beginning to appear — OpenClaw, which lives on your machine and can act from messaging apps to carry out

concrete tasks; Claude Cowork, which turns conversation into the real delegation of work on your desktop; the agents inside Microsoft Copilot and Copilot Studio, integrated into the enterprise ecosystem to automate processes across real data and applications. They are not tools for building. They are assistants that promise to do the work while you do something else.

But even those promises still have something of the frontier about them. Using AI in 2026 feels a lot like using the internet in 1997: you know there is something extraordinary in there, but getting to it still requires patience, trial and error, and the willingness to keep learning before you see results.

What is missing is the Google of AI. The interface that takes all that chaos — the technical frameworks, the promising agents, the models competing with one another — and turns it into a white search box on an almost empty page. The moment when artificial intelligence stops requiring technical knowledge to use well and becomes something anyone can benefit from without thinking about the models underneath, without understanding the difference between RAG and fine-tuning, without knowing what a token or a parameter is.

That moment will arrive. In some products, in some contexts, it is already arriving. And when it arrives fully — when AI becomes as invisible as Google — we will stop talking about it as if it were special.

It will become the air.

And then, as always, the generation that lived the before and after will be the only one that remembers what it took to get there. The one that remembers the chaos before order. The one that knows that behind every simple text box there are years of extraordinary work the end user will never see and never need to see.

That is the only real advantage of having arrived before the map.

It is not nostalgia. It is perspective.

And in technology, perspective is worth more than any certification.



*Google turned twenty-five in 2023. To mark the anniversary, it shared some statistics: more than eight billion searches a day. The most visited*

*site on the internet for two decades. Ninety-two out of every hundred searches in the world passing through its servers. What began in 1998 as an experiment by two students in a garage in Menlo Park became the invisible infrastructure through which the world finds almost everything it needs to find. That is how technological revolutions that truly work come to an end — not with a final explosion, but with a silent disappearance. They become so necessary that they stop being seen. In 2026, while AI still makes noise, generates headlines, and awakens fears and hopes in nearly equal measure, it is worth remembering that the noise is temporary. What remains after the noise is what really matters.*

## **ACT III — THE FRONTIER THAT BREATHES**

## Chapter 7 — The Second Threshold

There is a feeling that has been following me for the past couple of years.

It is not exactly an image. It is something harder to explain. The feeling of standing in a place I already know, even though I have never been there before. Of recognizing a territory at the very moment I am seeing it for the first time.

The first time I saw someone create an artificial intelligence agent from scratch — without writing code, without setting up infrastructure, without depending on a team of engineers behind them — in a matter of minutes and using only natural language, something lit up in my memory with a clarity that surprised me.

It was not just amazement.

It was recognition.

I had felt it before.

In a small shop with a few computers connected to something that still had no clear name. In front of a screen that took too long to load something arriving from somewhere else in the world. In the exact moment when a page appeared, slow, fragmented, real.

The same mix of fascination and vertigo. The same intuition that something important was happening before the right words existed to explain it. The same certainty of standing in front of a technology that had not yet fully shown what it was going to become, but had already started to change the relationship between people and machines.

The second threshold.



What is happening in 2026 with artificial intelligence does not quite resemble anything that came before. And yet, for those of us who lived through the birth of the internet, there is something deeply familiar about this moment.

It looked nothing like what I'd seen before. But it felt exactly like it.

The speed. The disorientation. The sense that tools are appearing faster than we can understand them. The number of people speaking with certainty about something that is still taking shape. The excess of enthusiasm. The excess of fear. And in the middle of all that, a group of people trying to learn on the fly because they understand, even if they cannot yet prove it, that this is not a passing trend or a technical curiosity.

That is how it was with the internet.

First it was a strange territory inhabited by very few. Then it became a promise. Then an explosion. And long before the rest of the world integrated it into everyday life, there was already a minority that could feel something big moving under the surface.

Artificial intelligence differs enormously from that revolution, of course. They are not equivalent technologies or identical processes. But they share something decisive: that initial moment when most people still do not really know what they have in front of them, while a few already sense that the world has tilted slightly — and that this tilt is going to change everything.



The first great revolution of the internet was access.

Access to information. To communication. To publishing. To a new way of crossing distance.

The revolution now taking shape is different. Stranger, even. Because it does not stop at expanding access. It begins to touch something else: action.

For decades, interacting with a computer meant telling it exactly what to do. Even when the interface became friendlier, even when the design improved, even when applications seemed more intelligent, the underlying logic remained the same: the machine waited for precise instructions. You gave the order. It executed.

With agents, that begins to change.

For the first time in a massive, visible, and relatively accessible way, we are facing systems that are not just asked to perform a specific task, but to pursue an objective. Something they interpret, break down, and push forward in a way that no longer feels like simple execution. It feels like something else. Something closer to collaboration, or at least to an early version of it.

That changes the experience.

And for those of us who came through the first revolution, there is something almost emotional in recognizing that shift. Not because we know exactly where it will take us, but because we know the smell of these stages. We know what a territory feels like before the map. We know the difference between a useful tool and a crack in history.



There is another reason this moment feels so much like the first one: the threshold has dropped again.

In the 1990s, connecting to the internet required patience, technical context, access, and time. Publishing something required even more. There were barriers. They were not insurmountable, but they were there. And each time the threshold dropped, the number of people able to participate expanded all at once.

That is happening again.

Today, a person without deep technical training can do things that, very recently, required programmers, budgets, infrastructure, or knowledge that was not within reach of just anyone. They can talk with an artificial intelligence, ask it to perform complex tasks, organize processes, generate content, automate parts of their work, explore ideas, test out solutions. They can even begin constructing behaviors that look a great deal like those of a real assistant.

That lowering of the threshold is one of the clearest signs that something important is happening.

Not because it guarantees that everything ahead will be good. But because it marks the moment when a technology stops belonging only to specialists and begins to enter — with all its discomfort and all its power — into the hands of ordinary people.

That is when a revolution stops being only a promise and starts becoming a problem, an opportunity, a debate, a habit. Reality.



There is a very particular moment in every technological revolution.

The moment when the machine stops feeling like an object and starts feeling like a presence.

I do not mean consciousness. I do not mean humanity. I do not mean any of that. I mean something else, more concrete and more recognizable: the moment when it stops being only a mute tool and begins to answer in a way that alters your perception of what is in front of you.

I felt that with the first computer-to-computer connections. With BBSs. With the web appearing on the other side of a slow screen. And I felt it again when I began genuinely interacting with artificial intelligence systems capable of sustaining a conversation, interpreting intention, continuing a line of thought, and returning something that did not stop at the literal instruction.

It was not magic. It was not science fiction. It was something else.

It was the return of a feeling I had believed was unrepeatable.

That is why I speak of a second threshold. Not because artificial intelligence is a repetition of the internet, but because it activates in some of us the same intimate alarm. The one that tells you this is not an incremental improvement. That it is not just one more tool in a long list of tools. That something has shifted.

And that if you do not pay attention now, later you will remember this moment the way these things are always remembered: too late, once the change has already become part of the landscape and nobody remembers very clearly how it began.



I do not know what exact form all of this will take.

I do not know which of the tools that seem central today will still exist in two years. I do not know which promises will be fulfilled and which will be left behind. I do not know whether, in a decade, we will look back on this as the real beginning or merely as an early and clumsy version of something much larger.

But I do know this: I recognize the pattern.

I recognize the texture of these moments. The mixture of noise and revelation. The difficulty of separating what is deep from what is superficial. The initial awkwardness.

The sudden acceleration. The intuition — still difficult to defend with airtight arguments — that something has already changed even though nobody can yet fully measure how much.

I lived through that once already.

And living through it a second time produces a strange emotion. A mixture of privilege and responsibility. Of enthusiasm and caution. Even of gratitude. Because not everyone gets to see, twice, the edge of what is possible move.



Some people arrive at these technologies as if they were completely new ground.

Others arrive carrying a memory with them.

Not a nostalgic memory. Not the empty idea that everything used to be better. A more useful memory than that: the memory of having seen a revolution start as confusion, then become exaggeration, then become inevitability. The memory of knowing that the noise at the beginning does not invalidate the depth of the change. The memory of understanding that the most important things rarely introduce themselves in an orderly way.

That memory does not give you certainty.

But it gives you perspective.

And in moments like this, perspective is worth more than it seems.



*In 1969, long before the internet boom many of us lived through, ARPANET connected four university computers in the United States for the first time. Nobody at that moment could imagine the web, social media, e-commerce, or conversational artificial intelligence. But something was already there: the first sign of a new relationship between machines, information, and distance. Technological revolutions almost never announce themselves as revolutions. At first they are only a small shift in behavior in some technical corner of the world. The rest comes later.*

## Chapter 8 — Where Control Ends

Every technological revolution has a hinge moment.

Not the moment when the technology appears — that is usually technical, quiet, reserved for those already on the inside. But the moment when it stops being a tool you use and begins operating with enough autonomy to change the way you work, decide, and delegate.

With the internet, that moment came when the network stopped being only a place of consultation and began organizing entire processes of everyday life. When it was no longer just for looking something up, but for buying, selling, speaking, publishing, managing, remembering, choosing. When it stopped being a place you visited and became infrastructure.

With artificial intelligence, that moment has another name.

Agentification.



Agentification is not a fad, or a company, or a specific tool. It is a shift in logic.

For decades, the relationship with computers worked under a relatively clear rule: you gave instructions and the machine obeyed. It could do so better or worse. Faster or slower. Through a friendlier interface or a rougher one. But the underlying logic did not change. The sequence began with you. Every action required an order. Every result depended on someone having defined beforehand the steps needed to get there.

An agent breaks that sequence.

Not because it thinks like a person, or because it has human will, but because you are no longer asking it only for a task. You describe a goal. You give it tools. You open a space for action. And within that space, it decides how to move forward.

That is what makes this moment so different from others.

We are not talking only about more efficient software. We are talking about systems capable of interpreting a general intention, breaking it down into steps, handling

intermediate obstacles, choosing possible paths, and producing outcomes without a person having to intervene in every movement.

That shift — from the precise instruction to the interpreted objective — is one of the most important transformations of this stage.



That is why the key word is not intelligence.

The key word is judgment.

Because an agent can be useful, fast, surprising, even brilliant in certain contexts. But all of that exists alongside something more uncomfortable: when the human instruction is incomplete — which is almost always the case — the agent does not necessarily stop to ask for clarification. It interprets. It fills in gaps. It moves forward with what it considers the most reasonable reading. And that is where the real problem of this era begins.

Not the simple science-fiction fear of a machine rebellion. Something far more grounded, and much closer.

The problem of systems that execute effectively against goals defined in ambiguous ways.

An agent does not need to fail spectacularly to create a problem. It only needs to misread an intention, fail to detect an edge case, optimize something it should not have optimized, or insist on fulfilling an objective without understanding the human context surrounding it.

That is where fascination becomes responsibility.



Most human instructions are blurry.

At work, in everyday life, even between people who know each other well, we constantly complete what the other person meant to say. We understand tone, infer priorities,

grasp implicit limits, make corrections in real time. A great deal of human communication works precisely because it is not necessary to explain everything.

With agents, that habit becomes risky.

Because the more autonomous the system is, the more it matters not only what you asked for, but what you left unsaid. What you assumed it would understand. What part of the context you thought was obvious. What edge of the problem you never bothered to define because a person would have known to stop on their own.

That is the great lesson of this stage: the quality of the result does not depend only on the model, or the tool, or the speed, or even access to data. It also depends on how clearly the space of action has been thought through.

It's not enough to say what you want.

You have to think about what happens when it takes you literally — and runs with it further than you meant.



That is where guardrails appear.

The word sounds technical, but at bottom it refers to something deeply human: limits. Restrictions. Frames. Instructions about what must not be done, in addition to what should be done.

For a long time, the main challenge in technology was getting machines to do more things. Today, in many cases, the challenge is beginning to become something else: making sure they do not do too much.

Making sure they do not insist where they should stop. That they do not invent context where none exists. That they do not choose a path acceptable in terms of efficiency but wrong in terms of judgment. That they do not confuse autonomy with unrestricted permission.

Designing an agent well does not consist only in making it capable.

It consists in making it reliable within a frame.

And that frame does not come from the model. It comes from the person who designs it, configures it, deploys it, or uses it.



That changes where responsibility lives.

For a long time, technological mistakes seemed to reside in code, infrastructure, or the wrong tool. And of course that still exists. But at this stage another layer of responsibility appears, one that for years was less visible: the responsibility of thinking clearly before delegating.

What exactly do you want to automate? What part of the process can actually be handled on its own, and what part requires human supervision? What information is that agent allowed to use? What decisions may it make? What decisions should it never make? What happens if it misreads an instruction? What happens if it encounters an exception? What happens if the case that appears does not resemble the ideal case it was designed for?

These questions are not an appendix to design. They are the design.

And we still are not teaching them with enough seriousness.



I have seen something like this before.

In the 1990s, while I was building networks and configuring servers, almost nobody talked about security as a central discipline. The internet was new, powerful, chaotic. The urgent thing was to get connected. The rest would come later. And later came exactly what you would expect: viruses, spam, phishing, identity theft, attacks, leaks, exposed infrastructure. Security became a priority when the problems were already underway.

Always running behind.

The difference is that now, at least, we have the possibility of recognizing the pattern earlier. We know that when a technology lowers the threshold, scales quickly, and starts entering real processes, problems do not take long to appear. We know that initial convenience often hides complexities that only become visible once thousands or millions of people start using the same thing in different ways.

History does not guarantee that we will act more wisely this time.

But at least we can no longer say we did not see the difficult part coming.



There is something else that makes this stage unique: speed.

The internet took years to normalize. Conversational artificial intelligence took months. What used to be absorbed over a decade can now change visibly from one quarter to the next. New tools. New capabilities. New interfaces. New promises. All at once.

That speed makes clarity even more important.

Because when a technology moves faster than culture's ability to understand it, the risk is not only technical. It is also human. Organizational. Political. Ethical. People start using things before they have thought through the place those things should occupy. Companies delegate before defining limits. Teams adopt before discussing judgment. And then the tool gets installed not because it is understood, but because it is already available.

That too is a form of design. It simply happens by omission.



That is why the frontier that breathes is not just an image.

It is a fairly literal description of this moment.

We are in a living territory, a shifting one, moving while we try to understand it. A frontier that does not wait for us to finish mapping it before continuing to advance. An unstable edge between what we still fully control and what we have already begun to delegate without fully understanding what that delegation means.

That is where we are.

Not in the future. Here.

And that is why it matters so much who is thinking through this stage. Not only the engineers building models. Also the technicians implementing them, the leaders

purchasing them, the teams using them, the organizations integrating them into real processes, the people deciding what level of autonomy they are willing to accept.

Because a technology is not defined only by what it can do.

It is also defined by the limits a society decides to place on it.



I do not have fixed answers to all of this.

I do not know what the right balance will be between autonomy and control. I do not know which of today's guardrails will look naïve in five years. I do not know which regulatory frameworks will survive and which will be obsolete before they are even implemented. I also do not know how much of the current conversation is real lucidity and how much is just the inevitable noise of the beginning.

But I do know one thing: this time the challenge is not only arriving early.

It is arriving early and thinking well.

It is not enough to be amazed. It is not enough to adopt. It is not enough to experiment. All of that matters, yes. But if this revolution is truly going to enter our decisions, our work, the circulation of information, and the way we delegate tasks and judgment, then the question is no longer only what this technology can do.

The question is what part of ourselves we are willing to hand over to it.



*The word "agent" in artificial intelligence has a long history. It has appeared in academic literature for decades, ever since researchers began imagining systems capable of acting autonomously in complex environments. For a long time it was more a theoretical category than a practical one. In 2026, it stopped being only that. Agents now live on personal laptops, inside productivity tools, in startups, in laboratories, and in enterprise environments. What was once a conceptual category has begun to turn into everyday experience. And when a technical idea enters*

*daily life, it is no longer enough to ask whether it works. You also have to ask under what conditions it should work.*

## Chapter 9 — The Cursor Still Blinks

There is an image I have carried with me since I was nine years old.

A blue screen. A blinking cursor. The silence of a dining room at night, with the only light coming from a television connected to a machine my father had repaired with his own hands.

I did not know then that that cursor would accompany me for the rest of my life.

I saw it blinking on the Commodore 64 while I copied lines of code from a magazine. I saw it blinking in a Linux terminal while I configured a server no one else in the office knew how to touch. I saw it turn into Netscape — no longer a cursor but a loading bar moving slowly, pixel by pixel, while a page drew itself from some server on another continent, fragmented and slow and completely extraordinary. I saw it in Google, in the late 1990s, when I typed something into that white search box and got what I was looking for faster than we were used to, faster than anything had ever answered before.

And I saw it blink again in 2023, in a white chat window, while an artificial intelligence processed my first real question — not a test, not an experiment, but a genuine question addressed to something I sensed could answer in a way nothing before it could.

The cursor blinked.

And answered.



Thirty-five years separate those two moments. Thirty-five years of technology accumulating in layers, of networks built on top of networks, of protocols speaking to protocols, of millions of people learning new languages in order to communicate with machines that were, at the same time, learning to communicate with them.

Thirty-five years that, in retrospect, look like preparation.

Not for AI in the abstract, but for this specific moment. The moment when the distance between human intention and the execution of that intention became so small it almost disappeared. The moment when writing what you want produces what you want, with a precision and a speed that five years ago would have seemed impossible and that, five years from now, will probably look primitive compared with what is coming.

That is what we are living through. Not the end of something. The beginning of something that still does not have a complete name.



The two great revolutions running through this book — the internet in the 1990s and AI in the 2020s — have more in common than they seem to at first glance, and they are also more different than they seem on second thought.

What they have in common is the human part.

In both cases there were people who arrived before the map. Who learned a new language without having a complete manual. Who explained to a skeptical world why something that seemed distant and complicated would become indispensable and ordinary. Who built inside chaos without knowing exactly what they were building, but with the certainty that it was worth doing.

In both cases there was wonder. Vertigo. The feeling of standing at the edge of something whose bottom you could not see.

And in both cases — this is what is not often said — there was fear. Not paralyzing fear, but the kind that forces you to pay attention. The kind that says: this is real, this matters, this is going to change things you do not yet know are going to change.

That fear is a compass. It points toward what deserves attention.



What distinguishes them is speed and agency.

The internet took years to reach most people. AI took months. The internet gave people access to information and communication. AI is giving people access to the capacity to act — agents that execute, automate, decide, learn. The scale of change is different. The pace is different. And the responsibility of those who arrive first is different too.

My father was an electronics engineer. He arrived before the map in his own era — the era of the first transistors, of electronics as a new discipline, of machines that first took up an entire room, then a table, then a pocket.

He did not teach me technology. He taught me something harder and more valuable: that machines are not the opposite of the human. That understanding how things work does not take you away from the world, but brings you closer to it in another way. That curiosity is a form of respect — for the world, for what you do not yet know, for what is coming even if you still cannot see it.

I learned that in front of a blue screen with a blinking cursor.

I am still learning it.



This book ends here, but the story does not.

The story continues in the moment when you, reading this now, decide what to do with what you know. If you are one of those who lived through the first threshold, you have something no one can buy or learn quickly: the perspective of contrast, the memory of chaos before order, the earned certainty that these unmapped territories can be crossed.

If you are arriving at the second threshold without having lived through the first, you have a different advantage: you are not carrying some of the limits the past imposes. You do not know that this was impossible ten years ago, and that allows you to imagine more freely what might be possible ten years from now.

In both cases, there is something to do now.

Not tomorrow. Not when the manual is complete. Not when someone else has solved the difficult problems and made it safe, predictable, and boring.

Now. While the territory is still wild. While the rules are still being written. While you can still be one of those who arrived before the map, and not one of those who arrived when the map was already printed and framed on the wall.

The train does not wait. It never has.

But this time — unlike the last time — you know the train exists. You know what it sounds like when it starts. You know what it feels like to watch it pull away without you.

And you know, because you lived it or because someone who did is telling you now, that getting on is worth it.

It always was.



Out there, somewhere in the world, there is a child in front of a screen.

They do not yet know what they have in their hands. They do not know that what they are looking at will shape part of what the world is going to be in thirty years. They only know that light calls to them in a way nothing else does. That the screen is asking something of them — to speak, to write, to ask, to explore.

The cursor blinks.

It's waiting.

Some people don't teach you the future. They place it in front of you and trust that you will know what to do.



THE END

## About the Author

Lucio Damián Moschella is Global Head of IT at Axxon and a technology consultant, with more than twenty-five years of experience in Microsoft solutions implementation, infrastructure architecture, and digital transformation.

Born in Argentina, he developed much of his career between Buenos Aires and Mexico City, where he now leads the operations of Axxon Consulting, a firm specializing in Microsoft Dynamics 365 and enterprise cloud solutions.

Certified in Azure, Dynamics 365, and multiple Microsoft technologies, he has supported organizations throughout Latin America and Europe in their technological modernization processes.

CONNECTED is his first book.

LinkedIn: <https://www.linkedin.com/in/lucio-damian-moschella-23b1b9a/>